Modicon TSX Momentum

Adaptador de bus para Ethernet

Manual del Usuario

870 USE 112 03

12/98

33000804.00



Breite: 178 mm
Höhe: 216 mm

Datos, figuras, modificaciones
Los datos y dibujos no representan compromiso alguno. Queda reservado efectuar las modificaciones derivadas del progreso técnico. En caso de que Ud. tenga sugerencias acerca de mejoras o modificaciones, o en caso de que Ud. descubra errores en la presente documentación, rogamos nos lo haga saber. Un formulario al respecto encuentra Ud. en la última página de esta documentación.

Cursos
Para profundizar los conocimientos del sistema, Schneider Automation ofrece los cursos correspondientes.

Hotline
Las direcciones de nuestro Centro de Soporte Técnico encuentra Ud. al final de esta documentación.

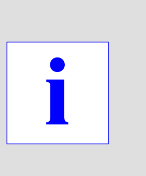
Denominaciones de productos
En general las denominaciones de productos de la Schneider Automation utilizadas en la presente documentación son denominaciones registradas por la mencionada firma.

Los nombres de productos restantes utilizados en esta documentación pueden también ser denominaciones registradas por otra empresas.
Microsoft y MS–DOS son denominaciones registradas y Windows es una denominación de la Microsoft Corporation en los Estados Unidos y en otros países.
IBM es una denominación registrada por la International Business Machines Corporation.
Intel es una denominación registrada por la Intel Corporation.

Contenido

Breite: 178 mm
Höhe: 216 mm

Información





Cuidado
Se deberán tomar en cuenta las siguientes reglas en aplicaciones con controles que deben cumplir con requisitos técnicos de seguridad.
Las reparaciones de componentes sólo deberán ser realizadas por el fabricante debido a razones de seguridad y mantenimiento de datos del sistema documentados.

Breite: 178 mm
Höhe: 216 mm

Símbolos utilizados



Nota
Este símbolo se utiliza para resaltar conceptos importantes.



Cuidado
Este símbolo advierte acerca de fuentes frecuentes de errores.



Advertencia
Este símbolo advierte acerca de fuentes de peligros que pueden conducir a danos financieros y de salud, o la posibilidad de consecuencias adversas.



Expertos
Este símbolo se utiliza para profundizar una información destinada exclusivamente a expertos (adiestramiento especial). Satear etsa información no tiene influencia alguna en la compresión de la documentación y no representa ninguna limitación en el uso standard del producto.



Consejo
Este símbolo sirve para resaltar la aclaración de sugerencias especialmente útiles en el manejo del producto.

Ejemplo: Este símbolo se utiliza para caracterizar ejemplos de aplicaciones.

▶ **Así se debe proceder**
Se indica aquí el comienzo de una serie de instrucciones, cuya ejecucíon es necessaria para lograr una función determinada del producto.

Los números se escriben de acuerdo al uso internacional a la representeción autorizada por SI (Système International d’ Unités). Los miles se separan mediante un espacio y se utiliza el punto decimal (por ejemplo 12 345.67).

Documentación adicional

Identificación	Número de pieza
Unidades de E/S para TSX Momentum, manual del usuario	870 USE 002 03
Ethernet TCP/IP Module User Guide (English)	840 USE 107 00
Modbus Protocol Reference Guide (English)	PI–MBUS–300

Conceptos y abreviaturas utilizados

El formato de escritura de números corresponde a la representación admitida según la práctica internacional y SI (Système International d' Unités): Los miles se separan mediante un espacio y se utiliza el punto decimal (por ejemplo 12 345.67).

TSX Momentum Adaptador de comunicaciones Ethernet 170 ENT 110 00

1

Se puede conectar este adaptador de comunicaciones a cualquier unidad de E/S TSX Momentum para crear un módulo de E/S funcional. Este provee la conexión directa con la red Ethernet, posibilitando a un host Ethernet la comunicación con dispositivos de campo cableados con las terminales de la unidad de E/S.

Figura 1 muestra el esquema de un adaptador típico y de una unidad de E/S.

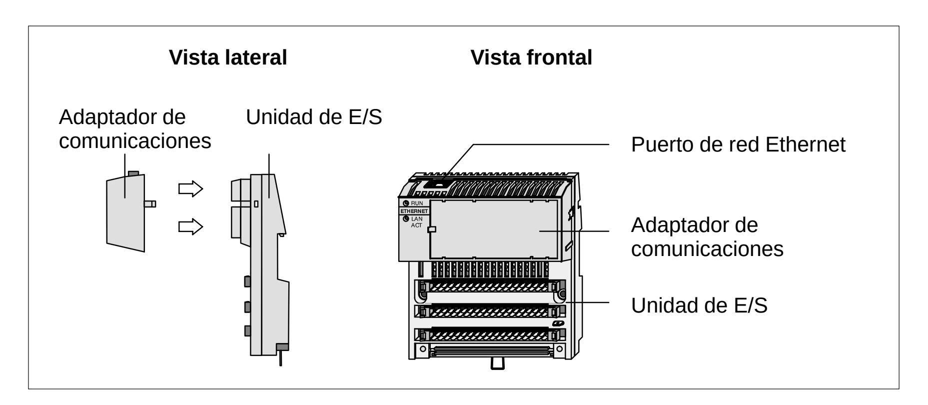


Figura 1 Adaptador de comunicaciones con unidad de E/S TSX Momentum

Este capítulo contiene lo siguiente:

- Descripción del producto
- Ejemplo: Tiempo de rotación de datos
- Indicadores de estado
- Conexión con la red
- Puesta en servicio del adaptador
- Reemplazo de una adaptador

Breite: 178 mm
Höhe: 216 mm

1.1 Vista general del producto

1.1.1 Función

Este adaptador se instala en cualquier unidad de E/S TSX Momentum para formar un módulo de E/S completo que comunica con una red Ethernet. Un controlador programable u otro dispositivo host en la red podrá leer desde las terminales de entrada y escribir en las terminales de salida de la unidad de E/S.

El adaptador comunica con dispositivos host usando el protocolo de aplicación Modbus con paquetes TCP/IP. Este apoya tanto Ethernet II como bloques de datos IEEE 802.3.



Para mayor información acerca del uso del protocolo de aplicación Modbus con TCP/IP, véase la Guía de Usuario para el Módulo Ethernet TCP/IP, número de material 840 USE 107 00. En la Guía de Referencia de Protocolo Modbus se proveen detalles acerca del protocolo Modbus, número de material PI–MBUS–300. Para mayor información acerca de la aplicación y cableado de campo de unidades de E/S, véase el Manual de Usuario TSX Momentum I/O Bases, número de material 870 USE 002 03.

1.1.2 Estructura física

Cada adaptador se conecta al conector de comunicación interno de su unidad de E/S. Afirma el adaptador en su sitio y se lo puede retirar liberadolo mediante un destornillador común. El usuario puede completar la etiqueta de cableado en el panel frontal (suministrada con la unidad de E/S) para identificar las conexiones del cableado con las terminales de la unidad de E/S.

El adaptador se considera como un equipamiento abierto y puede ser montado en una carcaza apropiada al sitio en donde se instala.

1.1.3 Tensiones de operación y control de error

El usuario provee la alimentación para el adaptador y unidad de E/S en el lugar del campo. El adaptador recibe su tensión de operación a través de su conexión interna con la unidad de E/S. El adaptador vigila su tensión y se separará de la red (offline) si la tensión no se encuentra dentro de la tolerancia.

1.1.4 Asignación de datos a las terminales de campo de la unidad de E/S

Los datos entre la aplicación y las terminales de campo de la unidad de E/S se asignan en el formato IEC. Véase el Manual de Usuario TSX Momentum I/O Bases, 870 USE 002 03 para las asignaciones de los diagramas para las unidades de E/S.

1.1.5 Administración del tiempo completo de reacción con las unidades de E/S

Para asegurar la determinación de tiempos de los mensajes de E/S, deberá Ud. diseñar la red para incluir solamente su host de aplicación y sus adaptadores de comunicaciones de la unidad de E/S. El agregado de otros tipos de dispositivos, como ser interfases de usuario o programadores, podrán ocasionar variaciones en los tiempos de mensajes de E/S, cuando estos dispositivos accedan a la red.

1.1.6 Especificaciones

Tabla 1 Especificación de red

Descripción	Especificación
Interfase Ethernet	Cumplimiento con la conexión STP ó UTP de 100 ohm.

Tabla 2 Agencias aprobadoras

Agencia	Estado
UL 508	Aprobación
CAN/CSA C22.2NO.142	Aprobación
CE Mark	Aprobación

Breite: 178 mm
Höhe: 216 mm

1.2 Ejemplo: Tiempo de rotación de datos

Figura 2 muestra un ejemplo de un bucle de control construido para medir el tiempo de movimiento de datos en las terminales de campo de un par de unidades de E/S.

Se conecta, a través de Ethernet, un ordenador central (host PC) ejecutando el programa de prueba, con dos adaptadores con unidades de E/S digitales. Las terminales de salida de campo de la unidad de salida están cableados directamente con las terminales de entrada de campo de la unidad de entrada. Se utiliza un osciloscopio para medir el tiempo de conmutación de las señales de campo.

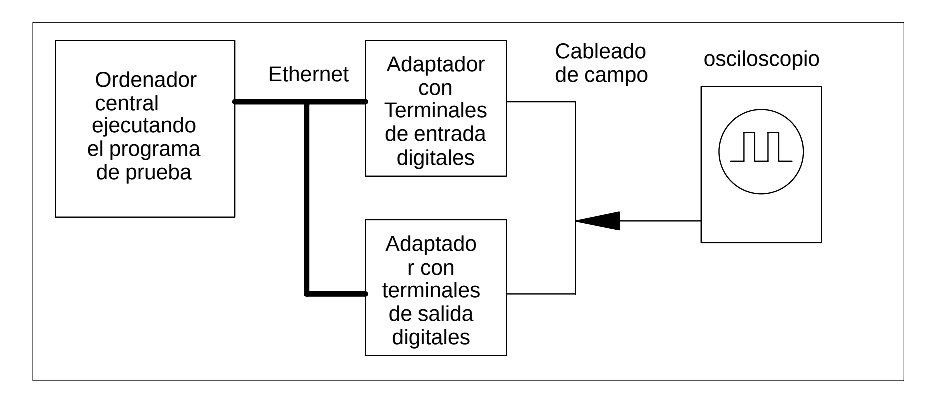


Figura 2 Ejemplo: Tiempo de rotación de datos

El programa de prueba en un bucle Java que ejecuta esta secuencia:

1. Lee continuamente las terminales de entrada del módulo de la unidad de entrada.
2. Escribe en la terminal de salida a una nueva condición (ON u OFF).
3. Cuando se recibe un estado cambiado desde las entradas, se alternan las salidas.

El osciloscopio mide el tiempo del estado ON de las salidas.

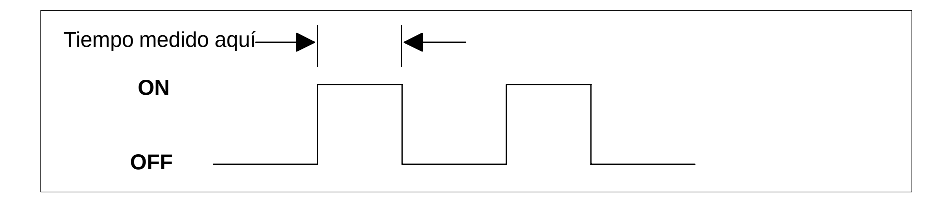


Figura 3 Medición del tiempo de rotación de datos

Las pruebas han sido efectuadas en dos estaciones de trabajo NT separadas, con estas configuraciones:

- 200 MHz, 96 MB RAM
- 100 MHz, 32 MB RAM

Tabla 3 muestra los tiempos de rotación de datos medidos. Los resultados indican que el factor que más afecta los tiempos de datos es la velocidad de la ejecución del bucle en el ordenador central.

Tabla 3 Resultados: Tiempo de rotación de datos

Dispositivos en la red	Carga de red	Tiempo mínimo	Tiempo máximo	Tiempo promedio	CPU central Velocidad y RAM
2	10%	5 ms	9 ms	6.2 ms	200 MHz 96 MB
2	40%	5 ms	9 ms	6.2 ms	200 MHz 96 MB
2	70%	6 ms	9 ms	6.3 ms	200 MHz 96 MB
64	10%	6 ms	8 ms	6.8 ms	200 MHz 96 MB
64	40%	6 ms	12 ms	8.4 ms	200 MHz 96 MB
64	70%	6 ms	13 ms	8.2 ms	200 MHz 96 MB
64	10%	25 ms	30 ms	26.7 ms	100 MHz 32 MB
64	40%	25 ms	30 ms	26.7 ms	100 MHz 32 MB
64	70%	26 ms	30 ms	27.0 ms	100 MHz 32 MB

Se reproduce en el capítulo 3 el código de fuente Java para el programa de prueba.

Breite: 178 mm
Höhe: 216 mm

1.3 Indicadores de estado

El adaptador posee dos indicadores en el panel frontal, que indican su estado operativo.

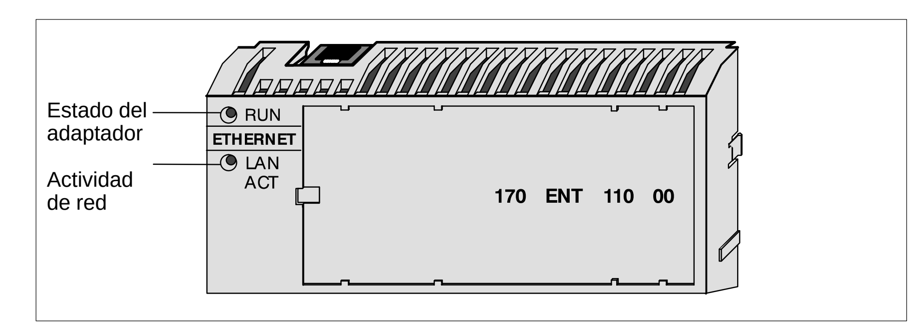


Figura 4 Indicadores

Tabla 4 Indicador RUN: Estado del adaptador

Estado del indicador	**Estado**
On (permanente)	Operación normal: existe alimentación desde la unidad de E/S y el adaptador se encuentra listo para la comunicación de red.
3 destellos, Off prolongado	Sin conexión: No se encuentra conectado el cable de red o está defectuoso.
4 destellos, Off prolongado	Sin dirección MAC: No se encuentra ajustada la dirección MAC. Problema interno de hardware.
5 destellos, Off prolongado	Si dirección IP: El adaptador está intentando obtener una dirección IP desde el servidor BOOTP.
6 destellos, Off prolongado	Se ha iniciado el programa ejecutivo interno del adaptador, pero no puede inicializar la unidad de E/S.
7 destellos, Off prolongado	El adaptador ha obtenido una dirección IP, pero no tiene un programa ejecutivo válido.
8 destellos, Off prolongado	Ha fallado el programa ejecutivo del adaptador durante la ejecución.
Destellando constantemente	El adaptador se encuentra cargando su programa ejecutivo.

Tabla 5 Indicador LAN ACT: Actividad de red

Estado del indicador	Estado
Destellando	Operación normal: El adaptador detecta actividad de red. La velocidad de destellos indica el monto de la actividad. Puede aparecer estacionariamente en On si la actividad de red es alta.
Off	El adaptador no se encuentra detectando ninguna actividad de red.

Breite: 178 mm
Höhe: 216 mm

1.4 Conexión con la red

1.4.1 Conector de red

El adaptador posee un conector RJ–45 para un cable 10Base–T UTP/STP (pares trenzados sin blindar o blindados). Se puede cablear el adaptador directamente con la unidad Ethernet.

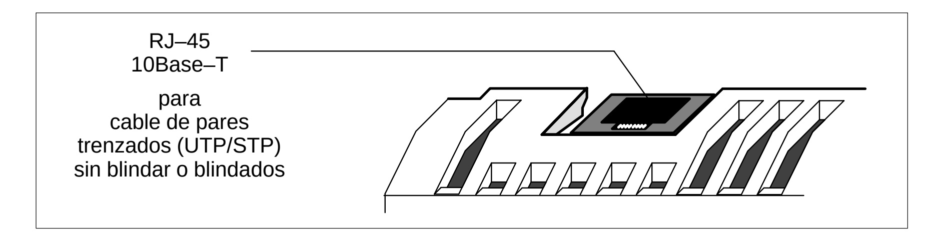


Figura 5 Conector de red

1.4.2 Etiquetas de red: Dirección Global y Dirección IP

El adaptador posee dos etiquetas montadas en sus paneles finales. Una etiqueta identifica la dirección global IEEE del adaptador (dirección MAC). La otra identifica su dirección de protocolo de Internet (dirección IP).

El instalador registra la dirección global y la entrega al administrador de red para su uso en el establecimiento de una dirección IP para el adaptador, durante el proceso BOOTP en el arranque. Una vez asignada la dirección IP, el administrado entrega esta dirección al instalador, que la escribe sobre la etiqueta de dirección IP del adaptador.

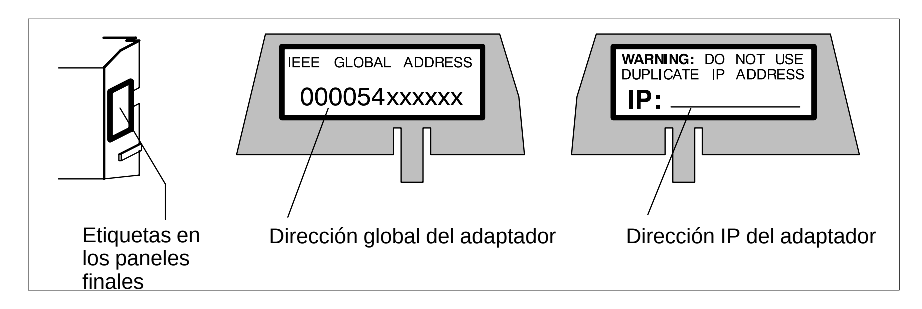


Figura 6 Etiquetas del adaptador: Dirección Global y Dirección IP

Breite: 178 mm
Höhe: 216 mm

1.5 Puesta en servicio del adaptador

1.5.1 Inicialización y autotests

Cuando el adaptador recibe su alimentación de operación inicial desde su unidad de E/S, ejecuta internamente la inicialización y los autotests. Si estos test fracasan, el indicador RUN destellará para indicar la razón de la avería, si es posible, y el adaptador permanecerá offline. Si los tests resultan exitosos, el adaptador intentará obtener su dirección IP Ethernet.

1.5.2 Asignar una dirección IP Ethernet

Vista general: Asignación de dirección
Se requiere un servidor BOOTP para asignar una nueva dirección IP al adaptador. Después de que el servidor asigna la dirección IP, la aplicación del servidor puede emitir un comando al adaptador para hacer que éste almacene la dirección internamente.

Si el adaptador ha almacenado su dirección y se vuelve a inicializar (por ejemplo, a continuación de una caída de la alimentación), el adaptador emitirá nuevamente el requerimiento para una dirección desde un servidor BOOTP. Si un servidor responde con una dirección, el adaptador usará esta dirección. Si el servidor no responde, el adaptador buscará su dirección almacenada.

Requerimiento de la dirección IP
Una vez completada su inicialización, el adaptador requerirá, desde un servidor BOOTP, su dirección IP Ethernet. El adaptador utiliza su dirección MAC con el protocolo BOOTP a lo largo de la red Ethernet.

Recibiendo la respuesta del servidor
El adaptador esperará diez segundos para que un servidor BOOTP responda con la dirección IP del adaptador. Si se recibe la respuesta del servidor, el adaptador

usará esa dirección mientras permanezca aplicada la alimentación a este adaptador.



Advertencia
PELIGRO POR DIRECCIONES DOBLES Si se tienen dos o más dispositivos con la misma dirección IP, podrá esto causar una operación no predecible en su red. Asegúrese que este dispositivo reciba una dirección IP única. La no observación de esta precaución puede conducir a perjuicios o daños en el equipamiento.

Reintentos hacia el servidor
Si no se recibe respuesta de un servidor BOOTP, el adaptador reintentará el requerimiento seis veces: tres veces utilizando el tipo de transferencia Ethernet II y tres veces utilizando el tipo de transferencia 802.3.

Sin recepción de respuesta del servidor (Dirección IP almacenada previamente)
Si el adaptador no recibe respuesta a ninguno de sus intentos por obtener una dirección IP, y si se ha almacenado previamente una dirección por intermedio de un comando de escritura Modbus desde la aplicación, el adaptador usará esta dirección almacenada.

Sin recepción de respuesta de servidor (Dirección IP no almacenada)
Si el adaptador no recibe respuesta a ninguno de sus intento para obtener una dirección IP, y no se ha almacenado ninguna dirección, el adaptador continuará intentando el requerimiento BOOTP cada 30 segundos. Durante este tiempo destellará su indicador RUN con el modelo ‘requiriendo’ (una secuencia de cinco destellos).

Breite: 178 mm
Höhe: 216 mm

1.5.3 Indentificar la unidad de E/S

Después de que el adaptador recibe su dirección IP, se ejecutará internamente un procedimiento para identificar su unidad de E/S. Si el procedimiento fracasa, el indicador RUN del adaptador destellará con un modelo de avería (seis destellos) y permanecerá offline.

Si se ha identificado exitosamente la unidad de E/S, el adaptador se encontrará listo para la comunicación usando el protocolo Modbus, a lo largo de TCP/IP.

1.5.4 Almacenamiento de la dirección IP en el adaptador

El adaptador posee un área RAM no volátil para el almacenado de su dirección IP asignada. Si la aplicación requiere que el adaptador retenga su dirección IP actual, la aplicación deberá emitir un comando de escritura Modbus para escribir un valor booleario en el registro específico en el adaptador, para hacer que sea almacenada la dirección. El estado predeterminado del adaptador es el de no almacenar la dirección.

La sección 2.1 describe como almacenar la dirección IP y como determinar si una dirección ya ha sido almacenada previamente.

1.6 Reemplazar un adaptador

1.6.1 Borrado de la dirección IP almacenada

Antes de retirar de servicio cualquier adaptador, deberá Ud. borrar su dirección IP.

El adaptador posee un área RAM no volátil para el almacenado de sus parámetros IP asignados. Los parámetros serán retenidos cuando se retire la alimentación del adaptador y se mantendrán permanentemente en el adaptador, si este es retirado del servicio. Si a continuación se vuelve a poner en servicio este adaptador, posiblemente ocasionará una actividad inesperada en su red. Ud. deberá por lo tanto, antes de sacar el adaptador de servicio, borrar los parámetros actuales.

El adaptador posee un registro interno que define el estado booleario (guardado y sin guardar) de sus parámetros IP. El registro puede ser leído por la aplicación y puede ser escrito para hacer que el adaptador borre sus parámetros.



Advertencia
PELIGRO POR DIRECCIONES DOBLES Si se tienen dos o más dispositivos con la misma dirección IP, podrá esto causar una operación no predecible en su red. Antes de retirar un adaptador de servicio, deberá Ud. escribir primeramente un 0 lógico (cero) en el registro de almacenamiento de parámetros, para borrar los parámetros almacenados. Esto reducirá la posibilidad de aparición de una dirección IP doble en su red, cuando el adaptador se restituya posteriormente en servicio. La no observación de esta precaución puede conducir a perjuicios o daños en el equipamiento. Véase la sección 2.1 para una descripción de los registros del adaptador, incluyendo la forma de borrado de los parámetros almacenados.

1.6.2 Retirar la alimentación de operación y desconectar el adaptador

Antes de retirar el adaptador, quítese la alimentación de operación de la unidad de E/S. Luego desconecte el cable Ethernet y retire el adaptador de la unidad.

Breite: 178 mm
Höhe: 216 mm

1.6.3 Instalar el nuevo adaptador

Móntese el nuevo adaptador sobre la unidad de E/S, siguiendo las instrucciones suministradas con este nuevo adaptador. Regístrese la nueva dirección global IEEE del nuevo adaptador (dirección MAC) y úsela para configurar una dirección de protocolo Internet (dirección IP) para el adaptador.

Las sección 1.5 describe la puesta en servicio del nuevo adaptador, usando el protocolo BOOTP Ethernet.

Comunicación con el Adaptador

2

- Registros de acceso de comunicación
- Registros de datos
- Registros de configuración
- Registros de estado

Breite: 178 mm
Höhe: 216 mm

2.1 Registros de acceso de comunicación

Cada adaptador contiene tres grupos de registros que posibilitan a la aplicación la comunicación con el módulo de unidad de E/S. La aplicación accede a los registros para transferir datos de entrada o salida a las terminales de campo del módulo de unidad de E/S, para ajustar o procurar la configuración del módulo, o para vigilar su estado.

Se puede acceder a todos los registros como referencias 4XXXX a través de los módulos de función MSTR en el programa de aplicación.

20

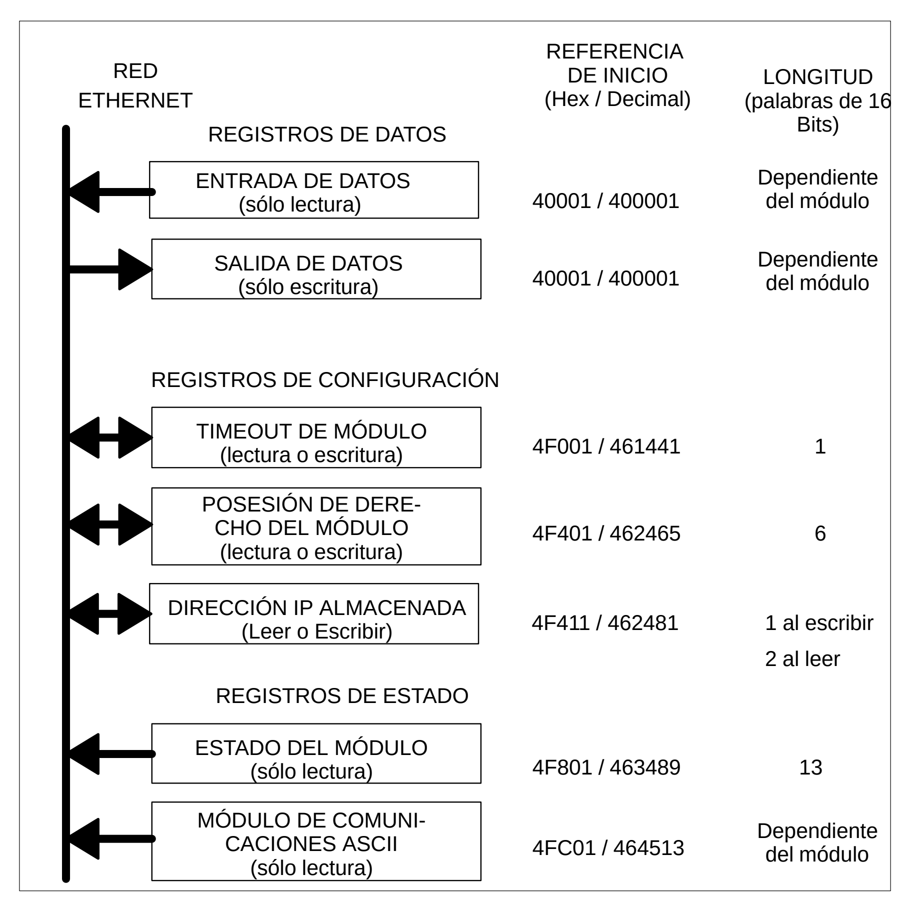


Figura 7 Registros de acceso del adaptador de comunicaciones

2.1.1 Registros de datos

40001 hexadecimal –– Entrada o salida de datos
Se utiliza la referencia inicial 40001 para direccionar datos de entrada provenientes de entradas de terminal y datos de salida a salidas de terminal. Los datos se transfieren en el formato IEC. Las asignaciones entre los registros de datos del controlador y las terminales de campo de la unidad de E/S es única en cada unidad, y se describe en el manual de usuario TSX Momentum I/O Bases, 870 USE 002 00.

Breite: 178 mm
Höhe: 216 mm

2.1.2 Registros de configuración

4F001 hexadecimal –– Valor del tiempo de retención de salidas
La referencia 4F001 especifica el tiempo en que las salidas quedarán retenidas en su estado actual, si no son actualizadas por intermedio de un nuevo comando de escritura Modbus. Si el tiempo de mantenimientothe del módulo expira antes de la recepción de un nuevo comando de escritura, se ajustarán todas las salidas a un 0 (cero) lógico.

La longitud del campo es de una palabra. El valor de tiempo de espera se expresa en unidades de 10 milisegundos con un valor de registro mínimo de 30 (300 milisegundos) y un valor máximo de 6000 (60 segundos). El valor predeterminado es 100 (1 segundo).

Se podrá leer el contenido del registro usando un comando de lectura Modbus.

4F401 hexadecimal –– Propietario del privilegio de escritura
Cuando el adaptador recibe por primera vez alimentación, le otorgará el privilegio de escritura solamente al primer participante que escribe en él, utilizando el comando de escritura Modbus. El adaptador posee un temporizador interno de 60 segundos para administrar el privilegio de escritura, y reservará solamente este privilegio a ese participante, mientras éste continúe escribiendo en el adaptador, dentro de intervalos de 60 segundos.

La referencia de inicio 4F401 especifica la dirección IP de hasta tres participantes más que pueden compartir su propio privilegio de escritura en el adaptador. Un participante que comparte el propio privilegio de escritura, puede escribir en el adaptador hasta tres direcciones IP (2 palabras por dirección), partiendo de 4F401. Con estas direcciones almacenadas, cualquiera de estos tres participantes podrá entonces escribir en el adaptador conjuntamente con el participante privilegiado originalmente. Esto permite que hasta cuatro participantes compartan su propio privilegio de escritura en el adaptador.

Si cualquiera de los tres participantes que cuentan con el privilegio de escritura escriben dentro del intervalo de 60 segundos, ningún otro participante podrá escribir en el adaptador. Si expira el tiempo del temporizador, cualquier participante podrá escribir en el adaptador.

Nótese que este temporizador para el privilegio de 60 segundos de escritura es independiente del temporizador de sostén de salidas y que sólo se aplica al privilegio de escritura. El período de tiempo de 60 segundos es un valor fijo. La aplicación no puede obtener acceso al mismo.

Cualquier participante podrá leer datos de entrada o información de estado del adaptador.

4F411 hex –– Dirección IP almacenada
Esta referencia sirve para un propósito doble, en dependencia de si la aplicación emite un comando de escritura Modbus o un comando de lectura Modbus.

Comando de escritura Modbus: Almacenar o borrar una dirección En caso de un comando de escritura Modbus, se tratará la referencia como un registro de una palabra, con la aplicación escribiendo una palabra de datos. Los datos de escritura Modbus pueden consistir de un 1 ó 0 (cero), que hace que el adaptador guarde o borre su dirección IP actual.

Si se escribe un dato 1 en la referencia, el adaptador almacenará su dirección IP asignada actualmente en su RAM no volátil. Si ocurre una nueva inicialización y el adaptador no puede encontrar un servidor BOOTP, el adaptador usará esta dirección almacenada.



Advertencia
PELIGRO POR DIRECCIONES DOBLES Si se tienen dos o más dispositivos con la misma dirección IP, podrá esto causar una operación no predecible en su red. Antes de retirar un adaptador de servicio, deberá Ud. escribir primeramente un 0 lógico (cero) en el registro 4F411, para borrar las direcciones almacenadas en el adaptador. Esto reducirá la posibilidad de aparición de una dirección IP doble en su red, si el adaptador se restituye posteriormente en servicio. La no observación de esta precaución puede conducir a perjuicios o daños en el equipamiento.



Advertencia
EL ADAPTADOR SE INICIALIZA CUANDO SE CAMBIA ESTE CONTENIDO Cualquier cambio en el estado del contenido de esta referencia ocasionará la reinicialización del adaptador.

Comando de lectura Modbus: Procurar dirección IP actual En caso de un comando de lectura Modbus, se tratará la referencia como un registro de dos palabras, con la aplicación leyendo dos palabras de datos. Si el adaptador posee parámetros IP almacenados en su RAM no volátil, retornará su dirección IP actual al comando de lectura Modbus, indicando de esta manera, que ha almacenado parámetros. Si no se encuentran actualmente almacenados parámetros IP, el adaptador emitirá para la lectura todos unos (FFFFFFFF hex).

Breite: 178 mm
Höhe: 216 mm

2.1.3 Registros de estado

4F801 hexadecimal –– Bloque de estado del módulo
Estos registros suministran información acerca del nivel de revisión del módulo y de los parámetros de operación actuales.

La longitud del bloque es de 13 palabras. La aplicación puede leer los registros, pero no pueden escribir en ellos.

Tabla 6 Esquema del bloque de estado del módulo

Referencia (hexadecimal)	Propósito	Contenido
4F801	Longitud del bloque de estado (palabras)	13 decimal
4F802	Cantidad de palabras de entrada de módulo de E/S	Dependiente del módulo
4F803	Cantidad de palabras de salida de módulo de E/S	Dependiente del módulo
4F804	Número de identificación del módulo de E/S	Dependiente del módulo
4F805	Adaptador de comunicaciones número de revisión	Formato: XR donde: X = 4 bits superiores, siempre 0000 R = 12 bits inferiores que definen la versión como 3 caracteres hexadecimales. Ejemplo: 100 hexadecimal = Ver. 1..00 200 hexadecimal = Ver. 2.00
4F806	Longitud del bloque de encabezamiento ASCII (palabras)	Dependiente del módulo
4F807	Última dirección IP a comunicarse con este adaptador en la transacción Modbus más reciente (palabra inferior de 2 palabras – véase 4F80D)	Dependiente de la dirección del participante
4F808	Tiempo de reserva restante del propietario	milisegundos
4F809	Tiempo de detención de salidas restante	milisegundos
4F80A	Estado de funcionamiento del módulo de E/S	8000 hexadecimal = bueno 0000 hexadecimal = malo
4F80B	Valor del último error del módulo de E/S	Dependiente del módulo
4F80C	Contador de errores del módulo de E/S	Total de errores 0000 ... FFFF hexadecimal
4F80D	Última dirección IP a comunicarse con este adaptador en la transacción Modbus más reciente (palabra superior de 2 palabras – véase 4F807)	Dependiente de la dirección del participante

Breite: 178 mm
Höhe: 216 mm

4FC01 hexadecimal –– Bloque de comunicación ASCII del módulo
Estos registros contienen una descripción de texto ASCII del módulo. La aplicación puede leer los registros, pero no puede escribir en ellos.

La longitud del bloque depende del tipo de la unidad de E/S a la cual se encuentra conectado el adaptador. La longitud máxima es de 64 bytes de caracteres ASCII, correspondiente a una longitud de 8 ... 32 palabras, como se especifica en la palabra 6 del bloque de estado del módulo (a partir de referencia 4F806).

El bloque contiene marcas para identificar las cantidades de palabras de entrada y salida y el código de identificación de la unidad de E/S. Ud. puede analizar el contenido del bloque a extraer esta información en su aplicación.

Tabla 7 ASCII Esquema de bloque de comunicaciones

Caracteres ASCII	Significado
`ETHERNET`	Identifica el adaptador de comunicaciones Ethernet
20 hexadecimal (32 decimal)	espacio
`IEC`	Se transfieren datos con unidad de E/S en formato IEC
20 hexadecimal (32 decimal)	espacio

`DIG:`
`EXP:`
`ANA:`
	Módulo digital (rango de identificación: XX00 ... XX7F hex)
	Módulo de experto (rango de identificación: XX80 ... XXBF hex)
	Módulo analógico (rango de identificación: XXC0 ... XXFE hex)

20 hexadecimal (32 decimal)	espacio
`inlen=`*n*	Palabras de entrada (*n* = cantidad de palabras, decimal)
20 hexadecimal (32 decimal)	espacio
`outlen=`*n*	Palabras de salida (*n* = cantidad de palabras, decimal)
20 hexadecimal (32 decimal)	espacio
`ID=0x`*nnnn*	Código de identificación de módulo (*nnnn* = código de identificación, hex)

Figura 8 muestra ejemplos de contenido de bloque de comunicaciones ASCII para dos unidades de E/S.

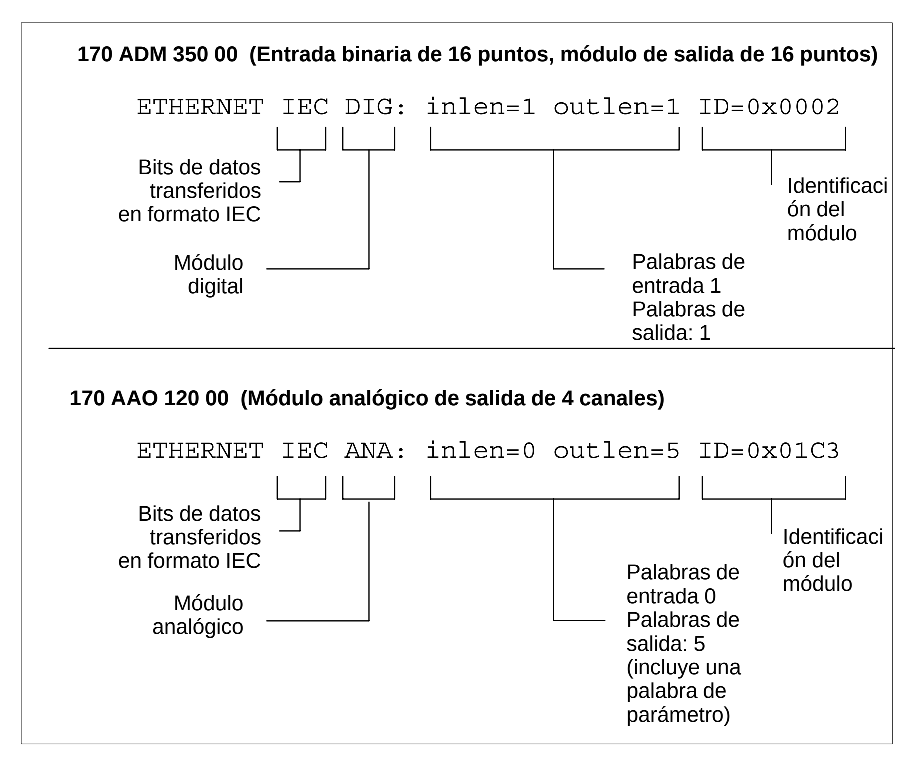


Figura 8 Ejemplos: Bloque de comunicaciones ASCII

Breite: 178 mm
Höhe: 216 mm

Programa de prueba: Código de fuente

3

- Fuente: response.java
- Fuente: test1.txt

Breite: 178 mm
Höhe: 216 mm

3.1 Fuente: response.java

Este programa Java organiza el bucle para la prueba de tiempos de reacción de datos, como se describe en el capítulo 1.

Este lee el fichero de texto `test1.txt` para procurar las direcciones de destino y los comandos Modbus para la comunicación con los adaptadores de comunicaciones de entrada y salida.

El contenido del fichero es como sigue:

```
// response.java, 6/17/98. This is a variation of mbtest.java.  This code
// will send a (write) command to one address and a separate command (read)
// to the second address. It will loop performing the read each time, but
// only doing the write when the read command detects a change in the input
// that was read. The loop counter is set as a constant. An improvement
// would be to pass it in the command line. The purpose of this program
// is to use an oscilloscope to measure the time elapsed between the input
// going high, and the output responding to it. The IP addresses of the IO
// modules, and the write and read commands are read from a text file.
//-------------------------------------------------------------------------

// mbtest.java 4/26/96

// minimal application to communicate and do performance measurement
// to the Quantum Ethernet module using ASA registered TCP port and
// MODBUS_ENCODING PDU format

// variant 5/14/96 to allow comparison of responses to the same request from 2 targets
// variant 5/24/96 to allow specification of independent addresses on the 2 targets
//                 and to allow time delay in script
//         11/11/97 generate 'usage:' and handle blank lines as comment

import java.io.* ;
import java.net.* ;

class mbtest {
   public static void main(String argv[]) {
      if (argv.length<1) {
         System.out.println("usage: java mbtest scriptfile > reportfile\n"+
                           "eg.  java mbtest script.txt > result.txt");
      } else
```

```
try {
   int dolog = 1;
   int keybuf;
   long skipcnt;
   int ioCount;
   byte expect;
   int first;
   DataInputStream di = new DataInputStream(new FileInputStream(argv[0]));
   String cmd;

   // analyse 'target' lines. Note must be followed by 'address'
   // note this version of the program requires that the 'target'
   // lines be the first 1 or 2 lines of the script

   String target;
   Socket es=null;
   OutputStream os=null;
   FilterInputStream is=null;
   cmd = di.readLine().trim();
   if (cmd.startsWith("target")) {
      target = cmd.substring(6).trim();
      System.out.println("mbtest: connecting to "+target);
      es = new Socket(target,502);
      os= es.getOutputStream();
      is = new BufferedInputStream(es.getInputStream());
      cmd = di.readLine().trim();
   }
   String target2;
   Socket es2=null;
   OutputStream os2=null;
   FilterInputStream is2=null;
   if (cmd.startsWith("target")) {
      target2 = cmd.substring(6).trim();
      System.out.println("mbtest: connecting to "+target2);
      es2 = new Socket(target2,502);
      os2= es2.getOutputStream();
      is2 = new BufferedInputStream(es2.getInputStream());
      cmd = di.readLine().trim();
   }
   int address = 1;
   int address2 = 1;
   byte obuf[] = new byte[261];
   byte obuf2[] = new byte[261];
```


Breite: 178 mm
Höhe: 216 mm

```
byte ibuf[] = new byte[261];
byte ibuf2[] = new byte[261];
obuf[0] = 0;
obuf[1] = 0;
obuf[2] = 0;
obuf[3] = 0;
obuf[4] = 0;
obuf2[0] = 0;
obuf2[1] = 0;
obuf2[2] = 0;
obuf2[3] = 0;
obuf2[4] = 0;
for (;;) {
   if (cmd.startsWith(”;”)) {
      System.out.println(cmd);
      cmd = di.readLine().trim();
      continue;
   }
   if (cmd.startsWith(”address2”)) {
      address2 = Integer.parseInt(cmd.substring(9));
      cmd = di.readLine().trim();
      continue;
   }
   if (cmd.startsWith(”address”)) {
      address = Integer.parseInt(cmd.substring(8));
      address2 = address;
      cmd = di.readLine().trim();
      continue;
   }
   if (cmd.startsWith(”quit”)) break;
   // handle script delay
   if (cmd.startsWith(”wait”)) {
      int delayTime = Integer.parseInt(cmd.substring(5));
      if (delayTime < 0 || delayTime > 30000) {
         System.out.println(”mbtest: warning: invalid delay time – ”+delayTime+”
– ignoring”);
      }
      else {
         Thread.sleep(delayTime);
      }
      cmd = di.readLine().trim();
      continue;
   }
```

```
if (os == null || os2 == null ) {
    System.out.println(”mbtestt: abort: no connection established to target”);
    break;
}

// now convert to a byte string
// assume format is hex separated by whitespace
int ix = 0; // output index = number of bytes found
int sx = 0; // input index = substring position
int l = cmd.length();
while (sx < l) {
    int ex;
    ex = cmd.indexOf(’ ’,sx);
    if (ex<0) ex = l;
    int bval = Integer.parseInt(cmd.substring(sx,ex), 16);
    obuf[7+ix++] = (byte) bval;
    sx = ex+1;
}

if (ix == 0) {
    // handle blank line as comment
    System.out.println(cmd);
    cmd = di.readLine().trim();
    continue;
}

obuf[5] = (byte) (ix + 1);
obuf[6] = (byte) address;
//-----------------------------------------------------------------
// read the command for the second target
//-----------------------------------------------------------------
cmd = di.readLine().trim();
ix = 0; // output index = number of bytes found
sx = 0; // input index = substring position
l = cmd.length();
while (sx < l) {
    int ex;
    ex = cmd.indexOf(’ ’,sx);
    if (ex<0) ex = l;
    int bval = Integer.parseInt(cmd.substring(sx,ex), 16);
    obuf2[7+ix++] = (byte) bval;
    sx = ex+1;
```


Breite: 178 mm
Höhe: 216 mm

```
            }

            if (ix == 0) {
               // handle blank line as comment
                System.out.println(cmd);
                cmd = di.readLine().trim();
                continue;
            }

            obuf2[5] = (byte) (ix + 1);
            obuf2[6] = (byte) address;
            cmd = di.readLine().trim();
         }  // end of FOR loop reading the input text
         // purge the key buffer
         //for( ; ; ) {
             //keybuf = System.in.available();
             //if( keybuf == 0 )
                 //break;
             //keybuf = System.in.read();
         //}
         keybuf = System.in.available();
         if( keybuf > 1 )  {
//             skipcnt = keybuf;
//             System.out.println("skip: "+keybuf );
             do  {
                 System.in.read();
                 keybuf--;
             } while( keybuf > 1 );
         }

         System.out.println("Press any key to stop");
         expect = 0x40;
         first = 1;
         // Now loop performing the IO
         for( ; ; ) { // loop doing IO
         //for( ioCount=0; ioCount < 500; ioCount++ ) { // loop doing IO
            int c = 0;
            int c2 = 0;
            int ix;
            keybuf = System.in.available();
            if( dolog == 1 )
              System.out.println("keybuf = "+keybuf );
            if( keybuf >= 1 )  {
```

```
    keybuf = System.in.read();
    break;
}
ix = (int)obuf[5] - 1;
os.write(obuf,0,ix+7);
c = is.read(ibuf,0,261);
if (c<=0) {
   System.out.println("mbtest: abort: detected unexpected close of channel");
   break;
}
while (c<7 || c<(6 + (0xff & (int)(ibuf[5])))) {
   System.out.println("mbtest: warning: response appears fragmented");
   int cx = is.read(ibuf,c,261-c);
   if (cx<=0) {
      System.out.println("mbtest: warning: response incomplete");
      break;
   }
   c += cx;
}


// verify input length against fragmentation
if (((int)(ibuf[5])&0xff) != (c-6)) {
   System.out.println("mbtest: warning: response length mismatch");
}
if( dolog == 1 )  {
  log_buf(obuf, ix+7, ">");
  log_buf(ibuf, c, "<");
}
// write to the second target, using the 2nd buffer
// wait until the input has changed
if ( first == 1 || ( ibuf[12] & 0x40 ) == expect ) {
    first = 0;
   ix = (int)obuf2[5] - 1;
   // if input bit 0x40 is on, turn on output bit 0x01
   if( expect == 0x40 )  {
       expect = 0;
       obuf2[16] = (byte) (obuf2[16] & 0xbf); // bit 40 off
       obuf2[16] = (byte) (obuf2[16] | 0x01); // bit 01 on
   }
   else  {
       expect = 0x40;
       obuf2[16] = (byte) (obuf2[16] | 0x40); // bit 40 on
```


Breite: 178 mm
Höhe: 216 mm

```
                obuf2[16] = (byte) (obuf2[16] & 0xfe); // bit 01 off
              }

              os2.write(obuf2,0,ix+7);

              c2 = is2.read(ibuf2,0,261);
              if (c2<=0) {
                 System.out.println("mbtest: abort: detected unexpected close of
channel");
                 break;
              }
              while (c2<7 || c2<(6 + (0xff & (int)(ibuf2[5])))) {
                 System.out.println("mbtest: warning: response appears fragmented");
                 int cx2 = is2.read(ibuf2,c2,261-c2);
                 if (cx2<=0) {
                    System.out.println("mbtest: warning: response incomplete");
                    break;
                 }
                 c2 += cx2;
              }
              if (((int)(ibuf2[5])&0xff) != (c2-6)) {
                 System.out.println("mbtest: warning: response length mismatch");
              }
              if( dolog == 1 )  {
                log_buf(obuf2, ix+7, ">");
                log_buf(ibuf2, c2, "<");
              }
              // check only the modbus response for equality (ignore address)
              //if (buf_diff(c, ibuf, c2, ibuf2)) {
                 //System.out.println("************* different");
              //}
           }
           if( dolog == 1 )
             System.out.println();
        } // end of FOR loop doing IO
     } catch (Exception e) { System.out.println("mbtest: unexpected exception:"+e); }
  }

  static boolean buf_diff(int c1, byte [] b1, int c2, byte [] b2) {
     // check only the modbus response for equality (ignore address)
     if (c1 != c2) return true;
     int i;
     for (i=7; i<c1; i++) {
```

```
            if (b1[i] != b2[i]) return true;
        }
        return false;
    }

    public static void log_buf(byte buf[], int len, String prefix) {
        System.out.print(prefix);
        int i;
        for (i=0; i<len; i++) {
            System.out.print(” ”+Integer.toString((buf[i]>>4)&0xf, 16)+
                                Integer.toString(buf[i]&0xf, 16));
        }
    System.out.println();
    }
}
```


Breite: 178 mm
Höhe: 216 mm

3.2 Fuente: test1.txt

Este fichero contiene las direcciones de destino de los adaptadores de comunicaciones de entrada y salida usados en el bucle de prueba. El servidor host tiene que resolver esto para las direcciones IP de los adaptadores.

El fichero contiene además los comandos Modbus de entrada y salida al adaptador para la lectura de las entradas del módulo de base y escribir las salidas del módulo de base.

El contenido del fichero es como sigue:

```
target eio4
target eio2
; input command followed by output command
3 0 0 0 2
10 0 0 0 2 4 00 00 00 40
quit
```